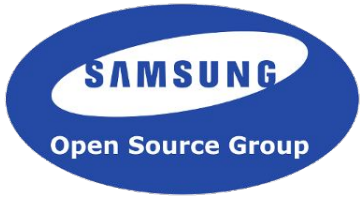
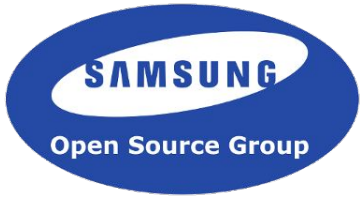


The state of Video Editing in GStreamer

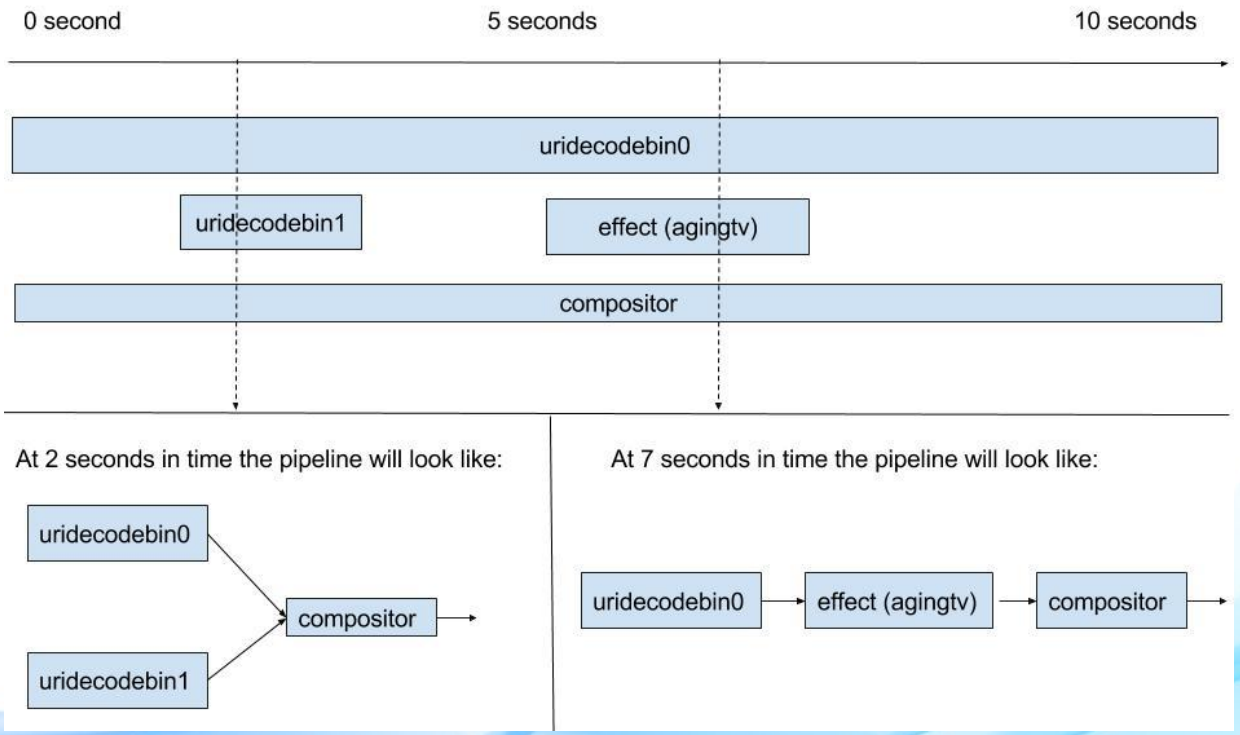
Thibault Saunier
Samsung Open Source Group
thibault.saunier@osg.samsung.com

History of multimedia editing with GStreamer

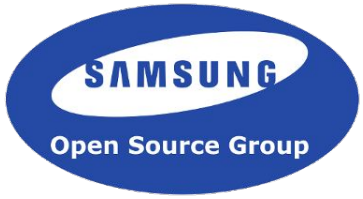


- GNonlin:
  - Started in 2001 by Wim Taymans
  - In 2004 Edward Hervey starts hacking on it when starting the Pitivi project
- GStreamer Editing Services:
  - Started in 2009 by Edward Hervey
- Non Linear Engine
  - Big refactoring of GNonLin in 2014

The NLE GStreamer plugin

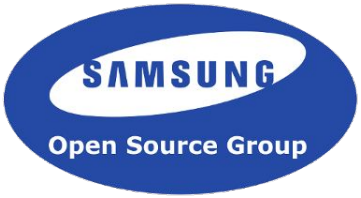


Major issues in GNonLin



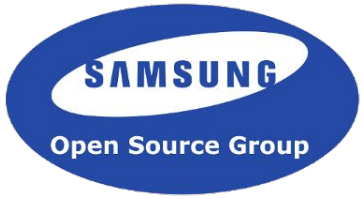
- Lack of thread safety
- **Many** useless threads were created

Lack of thread safety in GNonLin



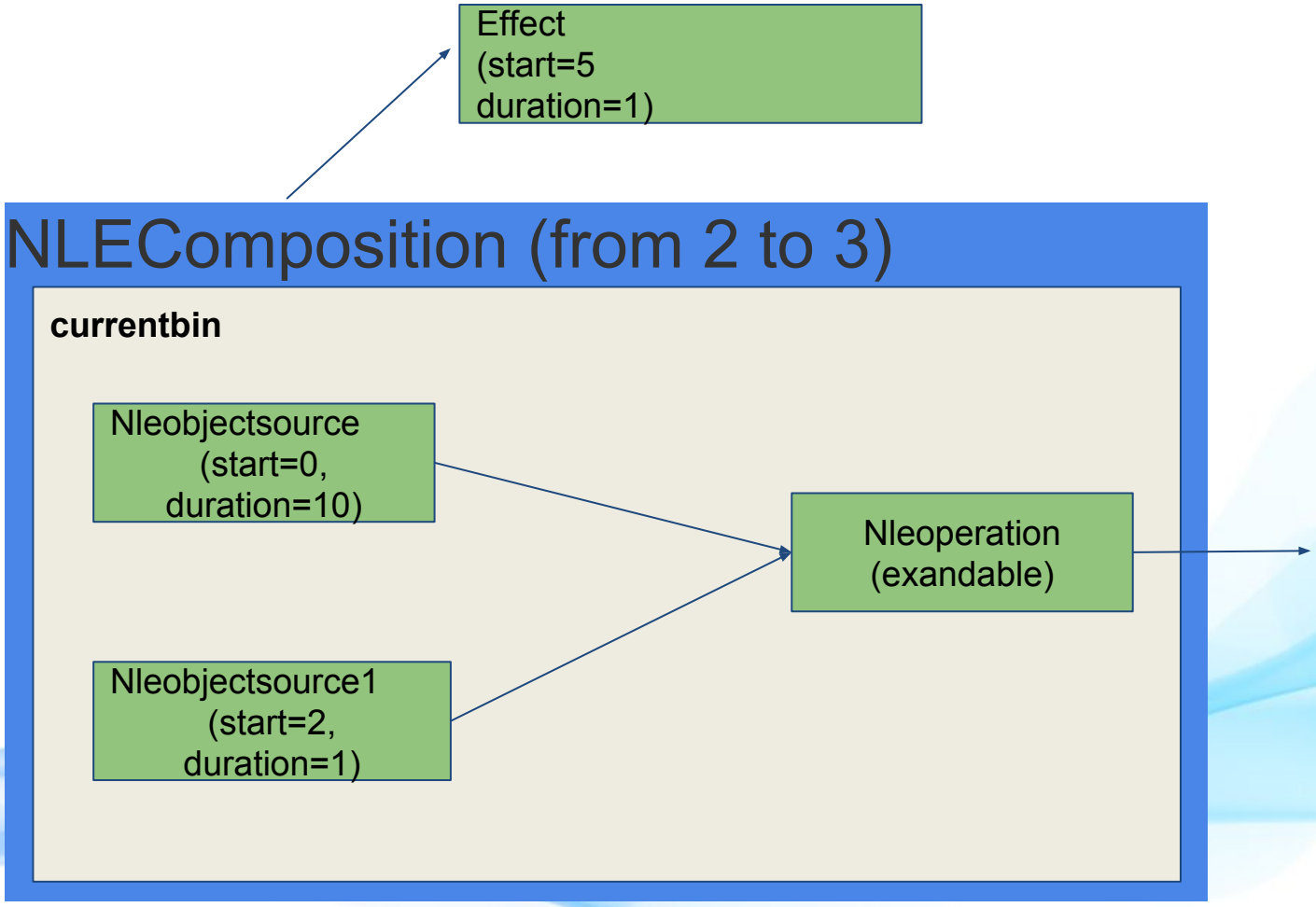
- gnlcomposition used to unlink/relink new pipelines from the streaming thread (where the EOS was received) or from the seeking thread
- In nlecomposition we introduced a new master thread where all the operations on children happen sequentially

Creation of many useless thread

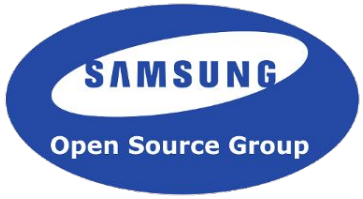


- In GnlComposition we used to have all elements inside in PAUSED state → many thread created and just waiting
- In NleComposition elements are not inside it until they are actually needed

nlecomposition

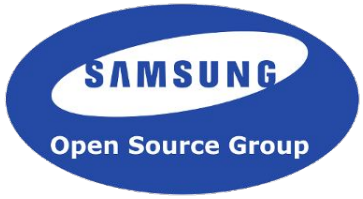


State of the Non Linear Engine



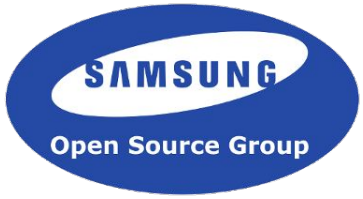
- Still lack testing outside GES
- Some part of the code could use refactoring
- Still lacking gst-launch support
- Most races are gone
- Working properly when used with GES

The GStreamer Editing Services



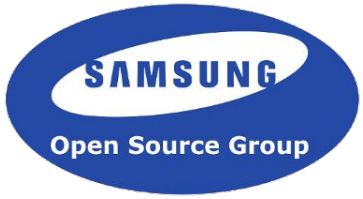
- Allow simple editing to get done as easily as possible
- Allow full fledged Video editing app to be built 'simply'
- **NON** goal of GES: handle all use cases where NLE is useful

Main features



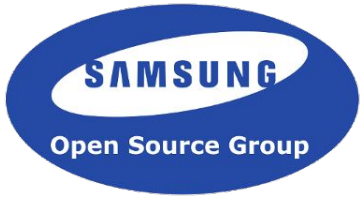
- Create a timeline, add clips, titles, transitions, effects, group them... easily
- GstPipeline subclass to play and render a timeline
- Edit the clips with advanced methods (ripple, roll, etc...)
- Asset handling support
- ges-launch-1.0 a tool to use GES on the command line

ges-launch got rewriten



- New 'command line formatter'
- Pretty flexible syntax that allows to virtually define any timeline

ges-launch examples



- Playback a portion of a media file

```
ges-launch-1.0 +clip /path/to/media inpoint=4.0 duration=2.0 start=4.0
```

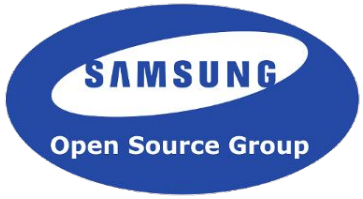
- Render two parts of two media files stitched

```
ges-launch-1.0 +clip /path/to/media i=4.0 d=2.0 \
   +clip /path/to/media1 i=5.0 s=2.0 d=1.0 \
   -f 'video/webm:video/x-vp8:audio/x-vorbis'  -o /path/to/rendered/file.webm
```

- Complex timeline

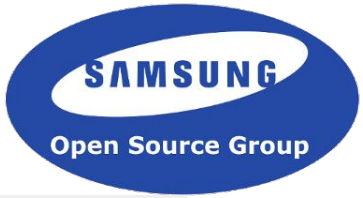
```
ges-launch-1.0 +clip /home/thiblahute/Videos/1.webm i=5.0 d=5 l=0 \
     set-posy 300 set-with 400 set-height 300 \
  +clip /home/thiblahute/Videos/2.webm s=0 i=20 d=5 l=2 \
    set-posx 150 set-posy 30 set-width 600 set-height 400
```

GstValidate integration



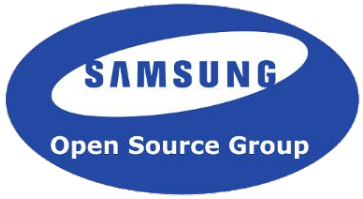
- GstValidateScenario: straight forward files to make actions on a pipeline, or any component
- Use to easily reproduce bugs (those scenario files are generated by Pitivi)

Scenario example



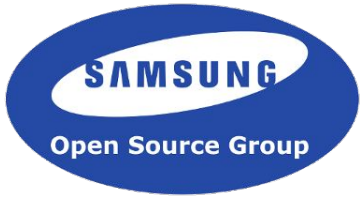

```
add-clip, name=c0, layer-priority=0,
  asset-id=file:///some/video, type=GESUriClip, \
  start=0, inpoint=0, duration=5.0

# Adding a clip to a second layer
add-clip, name=c1, layer-priority=1,\
  asset-id=file:///other/video, type=GESUriClip, \
  start=0, inpoint=0, duration=5.0

# Trim the clip
edit-container, edge=edge_start, container-name=c0, \
  position=1.0, edit-mode=(string)edit_trim;
```

Testing



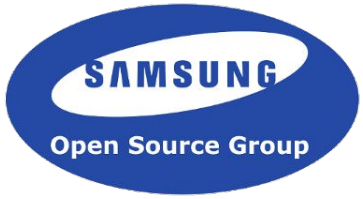
- Quite many unit tests (including python ones)
- Intensive integration testsuite running on ci.gstreamer.net playing, seeking and rendering timelines using GstValidate
- More tests to come, especially on plain NLE

Future of NLE



- It basically does what we want it to do
- Some code refactoring
  - No big plan at this point
- Maintenance / bug fixing

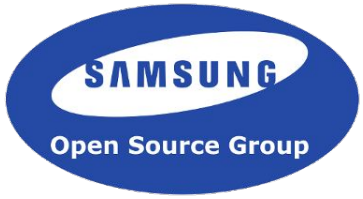
Future of GES



- Short/mid term
  - Refactor big chunk of it
  - Totally remove the concept of priorities

- Long term plan
  - Integration with higher level APIs (GstPlayer, GstTranscoding)
  - GES 2.0: the API could be drastically simplified and reduced

Thank you!



- Any question?